Deploy

CI/CD

Branch Protection

The master branch of the project is projected from direct changes. Changes can only be merged into the master branch, and then only when certain conditions are met.

Feature Branch Triggers

Code can be pushed to a feature branch without consequence. Raising a pull request should do the following:

The PR name is linted against a Jira ticket.

The code is compiled.

The code is subject to static analysis for security purposes.

The unit and integration tests are run.We only run in-process tests here. Some integration tests may be run later.

Code coverage is collected

A new artifact is created for this build.

*A new resource group is created in the cloud and the artifact + any requisite data is deployed there using IaC.

*The system tests are run against the newly created environment.Integration tests that require an environment will be run here.

*Code coverage is collected

* Branch-level deployment is an advanced strategy and a good target to aim for.

Pushing to a feature branch with an outstanding Pull Request re-runs the above checks.If branch-level development is present the testing environment can be safely torn down.

Master Branch Triggers

The code can be merged to the master branch with at least 1 acceptance in the PR process. Merging into master triggers the same actions as above but with deployment to UAT

Release Management

Creating a new tag for a commit on master that matches the semver pattern triggers several sequential actions:

The Staging environment for this project is updated with the infrastructure code from the tagged commit.

The artifact that was created for the tagged commit is deployed to Staging. This process can be triggered manually by specifying the commit to deploy, which facilitates a rollback if we deploy malfunctioning code.

Infrastructure as Code

Our infrastructure is written in declarative yaml, it is executed as part of our pipeline and stored in source control.

Monitoring

Logging in the Cloud

In a cloud environment it can be very challenging to debug software. We should implement detailed logging across the system, both to monitor healthy activity and to shine a light on errors. We will utilise the cloud provider's native logging services for this

Health Checks

Health checks ensure the system is up and responding, and may be used to verify specific system functionalities.

Alerting

If an error occurs, or a health check fails, an alert should be triggered to notify the relevant team members as soon as possible